Optimal coded caching in heterogeneous networks with uncoded prefetching

Emanuele Parrinello, Ayşe Ünsal and Petros Elia

***Abstract*— In the context of caching in heterogeneous networks, the work explores the setting where a multi-antenna transmitter (N_0 antennas), broadcasts to K receiving users, each assisted by one of $\Lambda \leq K$ helper nodes serving as limited-sized caches. Our aim is to identify the limits of coded caching when there are fewer caches than users ($\Lambda < K$), the interplay between having fewer caches but more transmit antennas, and the impact of non-uniformity where some caches serve more users than others.**

For a broad range of parameters, under the assumption of uncoded cache placement, the work derives the exact optimal worst-case delivery time (or equivalently, the optimal sum degrees of freedom (DoF)), as a function of the cache sizes and the user-to-cache association profile. This is achieved by presenting an information-theoretic outer bound based on index coding that adapts to user-to-cache association non-uniformities, and an optimal caching-and-delivery scheme. The result reveals the effect of these non-uniformities, and also reveals a powerful effect of introducing a modest number of antennas and a modest number of helper nodes; when $\Lambda < K/N_0$, adding a single degree of cache-redundancy yields a caching-gain increase equal to N_0, and similarly, adding antennas has a multiplicative DoF impact where for example introducing a second transmit antenna can double the DoF.

I. Introduction

A recent information theoretic exposition of the cache-aided communication problem [2], has revealed the potential of caching in allowing for the elusive scaling of networks, where a limited amount of (bandwidth and time) resources can conceivably suffice to serve an ever increasing number of users. This exposition in [2] considered a single-stream broadcast channel (BC) scenario where a single-antenna transmitter has access to a library of N files, and serves (via a single bottleneck link) K receivers, each having a cache of size equal to the size of M files.

In a normalized setting where the link has capacity 1 file per unit of time, the work in [2] showed that any set of K simultaneous requests can be served with normalized delay (worst-case completion time) which is at most $T = \frac{K(1-\gamma)}{1+K\gamma}$ where $\gamma \triangleq \frac{M}{N}$ denotes the normalized cache size. This implied an ability to treat $K\gamma + 1$ users at a time; a number that is often referred to as the cache-aided sum *degrees of freedom* (DoF) $d_\Sigma \triangleq \frac{K(1-\gamma)}{T}$, corresponding to a caching gain of $K\gamma$ additional served users due to caching.

For this same single-bottleneck setting, this performance was shown to be approximately optimal (cf. [2]), and under the basic assumption of uncoded cache placement where caches store uncoded content from the library, it was shown to be exactly optimal (cf. [3], [4]). This coded caching was adapted for a variety of basic broadcast settings that include uneven topologies [5], the erasure channel [6], [7], the MISO BC with fading [8], see also [9]–[12].

Cache-aided heterogeneous networks: A next step was to explore coded caching in the context of more involved topologies, that better capture aspects of wireless networks. Of particular interest are the so called heterogeneous networks where communication between the base station and the receiving nodes, takes place in the presence of helper nodes which can now serve as caches. This heterogeneous topology nicely captures an evolution into denser networks where many wireless access points work in conjunction with bigger base stations, in order to better handle interference and (when storage of data is allowed) in order to alleviate the backhaul load by replacing backhaul capacity with storage capacity at the communicating nodes.

The use of caching in such networks was famously explored in the *Femtocaching* work in [14], where wireless receivers are assisted by helper nodes of a limited cache size, whose main role was to bring content closer to the users. A transition to coded caching can be found in [15] which considered a similar heterogeneous network, where one receiving user can have access to a main base station (server) as well as to *multiple* access points (multiple helper caches). In this context, under mostly a uniform user-to-cache association, [15] proposes a coded caching scheme which is shown to perform to within a certain constant factor from the optimal. This uniform setting is addressed also in [16], again for the single antenna case.

Notation: For n a positive integer, we will use $[n] \triangleq \{1, 2, ..., n\}$, and we will use $2^{[n]}$ to denote the power set of $[n]$. The expressions $\alpha|\beta$ denote that integer α divides integer β. We will use $P(n,k) \triangleq \frac{n!}{(n-k)!}$ and $\binom{n}{k} \triangleq \frac{n!}{(n-k)!k!}$. If $\mathcal{A}$ is a set, then $|\mathcal{A}|$ will denote its cardinality. $\mathbb{N}$ will represent the natural numbers. We will use $Conv(f(i))$ to represent the lower convex envelope of the points $\{(i, f(i))|i \in [n] \cup \{0\}\}$ for some $n \in \mathbb{N}$. For $n \in \mathbb{N}$, we will use S_n to denote the symmetric group of all permutations of $[n]$. To simplify notation, we will also use such permutations $\pi \in S_n$ on vectors $\boldsymbol{v} \in \mathbb{R}^n$, where $\pi(\boldsymbol{v})$ will now represent the action of the permutation matrix defined by π, meaning that the first element of $\pi(\boldsymbol{v})$ is $\boldsymbol{v}_{\pi(1)}$ (the $\pi(1)$ entry of $\boldsymbol{v}$), the second is $\boldsymbol{v}_{\pi(2)}$, and so on. $\pi_s(\boldsymbol{L})$ will denote the sorted version of $\boldsymbol{L}$ in descending order.

II. System Model

In this work, we consider a basic broadcast configuration with a transmitting server having N_0 transmitting antennas and access to a library of N files $W^1, ..., W^N$ of total size $\sum_{n\in[N]} |W^n| = N$ units of 'file', where this transmitter is connected via a broadcast link to K receiving users and to $\Lambda \leq K$ helper nodes that will serve as caches which store content from the library. The communication process is split into a) the cache-placement phase, b) the user-to-cache

The authors are with the Communication Systems Department at EURECOM, Sophia Antipolis, 06410, France (email: parrinel@eurecom.fr, unsal@eurecom.fr, elia@eurecom.fr). This work was supported by the European Research Council under the EU Horizon 2020 research and innovation program / ERC grant agreement no. 725929.

Since its acceptance, this work has been extended and improved in its final version [1].

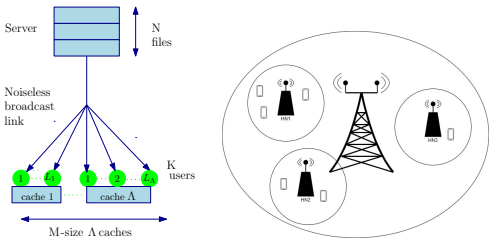

Fig. 1: Representation of the addressed problem (left) and a system example with $N_0 = 2, K = 6, \Lambda = 3$ (right)

assignment phase during which each user is associated to a single cache, and c) the delivery phase.

a) Uncoded cache placement phase: During this phase, helper nodes store content from the library without having knowledge of the users' requests. Each helper cache has size $M \leq N$ units of file, and no coding is applied to the content stored at the helper caches; this corresponds to the common case of *uncoded cache placement*. We denote by $\mathcal{Z}_\lambda$ the cache content at helper node $\lambda \in [\Lambda]$.

b) User-cache association: After the caches are filled, each user is randomly assigned to exactly *one* helper node/cache, from which it can download content at zero cost. We denote by $\mathcal{U}_\lambda$ the set of users associated to helper node/cache $\lambda \in [\Lambda]$. The user-to-cache assignment is independent of the cache content and independent of the file requests to follow. The resulting user-to-cache association is described by

$$\mathbf{L} = (L_1, ..., L_\Lambda)$$

where L_λ is the number of users associated to helper node/cache λ. Naturally $\sum_{\lambda\in[\Lambda]} L_\lambda = K$.

c) Delivery: The delivery phase commences when each user $k = 1, \dots, K$ requests from the transmitter, any *one* file W^{d_k}, $d_k \in [N]$ out of the N library files. Upon notification of the entire demand vector $\mathbf{d} = (d_1, d_2, \cdots, d_K)$, the transmitter aims to deliver the requested files, each to their intended receiver, and the aim is to design a *caching and delivery scheme* χ that does so with limited (delivery phase) duration T. For each transmission, the received signals at user k, take the form

$$y_k = \boldsymbol{h}_k^T \boldsymbol{x} + w_k, \quad k = 1, \dots, K \tag{1}$$

where $\boldsymbol{x} \in \mathbb{C}^{N_0 \times 1}$ denotes the transmitted vector satisfying a power constraint $\mathbb{E}(||\boldsymbol{x}||^2) \leq P$, where $\boldsymbol{h}_k \in \mathbb{C}^{N_0 \times 1}$ denotes the channel of user k, and where w_k represents unit-power AWGN noise at receiver k. We will assume that P is high (high SNR), we will assume perfect channel state information throughout the (active) nodes, statistically symmetric fading, and that each link (one antenna to one receiver) has capacity $\log(SNR) + o(log(SNR))$.

d) Performance measures: As in [2], T is the number of time slots, per file served per user, needed to complete delivery of any request vector[1]. We use $T(\boldsymbol{L}, \boldsymbol{d}, \chi)$ to define the delay required by some caching-and-delivery scheme χ to satisfy demand $\boldsymbol{d}$ in the presence of a user-to-cache association vector $\boldsymbol{L}$. Our interest is in the regime of $N \geq K$ where there are more files than users.

[1]The time scale is normalized such that one time slot corresponds to the optimal amount of time needed to send a file from a single-antenna transmitter to a single-antenna receiver, had there been no caching and no interference.

III. Main Results

We first describe the main results for the single antenna case, and then generalize to the multi-antenna case. The outer bound encompasses the class of all caching-and-delivery schemes χ that employ uncoded cache placement under a general sum cache constraint $\frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda} |\mathcal{Z}_\lambda| = M$ which does not *necessarily* impose an individual cache size constraint. The outer bound also encompasses all scenarios that involve a library of size $\sum_{n\in[N]} |W^n| = N$ but where the file sizes may be of different size. In the end, even though the actual designed optimal scheme will consider an individual cache size M and equal file sizes, the outer bound guarantees that there cannot exist a scheme (even in settings with uneven cache sizes or uneven file sizes) that exceeds the optimal performance identified here.

For the single transmit antenna case, the optimal delivery time takes the following form.

Theorem 1. *In the K-user, single-antenna network with Λ caches and cache-size γ, the optimal average delivery time $T^* \triangleq \min_\chi \mathbb{E}_{\boldsymbol{L}} \max_{\boldsymbol{d}} T(\boldsymbol{L}, \boldsymbol{d}, \chi)$ is*

$$T^* = \mathbb{E}_{\boldsymbol{L}}\left[Conv\left(\frac{\sum_{r=1}^{\Lambda-\Lambda\gamma} L_{\pi_s(r)}\binom{\Lambda-r}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}}\right)\right] \tag{2}$$

for points $\gamma \in \{\frac{1}{\Lambda}, \frac{2}{\Lambda}, \dots, 1\}$, where π_s is the permutation that sorts $\boldsymbol{L}$ in descending order.

The lower bound proof is in Section IV, and due to lack of space, the optimal scheme χ is presented in [17][2].

Effect of user-to-cache association non-uniformity: One can see that the different $\boldsymbol{L}$ can be split into classes

$$S_{\boldsymbol{L}} \triangleq \{\pi(\boldsymbol{L}) : \forall \pi \in \mathcal{S}_\Lambda\}$$

where each class defines a different type of non-uniformity in user-to-cache association[3]. To capture this effect of non-uniformity, we will here describe the optimal average delay for each type, by calculating

$$T^*(S_{\boldsymbol{L}}) \triangleq \min_\chi \mathbb{E}_{\boldsymbol{L}\in S_{\boldsymbol{L}}} \max_{\boldsymbol{d}} T(\boldsymbol{L}, \boldsymbol{d}, \chi)$$

representing the optimal delivery time averaged over the $\boldsymbol{L}$ of a given type.

Corollary 1. *Within any class $S_{\boldsymbol{L}}$, the optimal $T^*(S_{\boldsymbol{L}})$ takes the form*

$$T^*(S_{\boldsymbol{L}}) = Conv\left(\frac{\sum_{r=1}^{\Lambda-\Lambda\gamma} L_{\pi_s(r)}\binom{\Lambda-r}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}}\right) \tag{3}$$

at points $\gamma \in \{\frac{1}{\Lambda}, \frac{2}{\Lambda}, \dots, 1\}$

The lower bound proof is in Section IV, and the scheme is in [17].

We proceed to extend the above to the multiple antenna case.

[2]For any given $\boldsymbol{L}$, and any $\gamma \in \{\frac{1}{\Lambda}, \frac{2}{\Lambda}, \dots, 1\}$ the scheme achieves a worst-case delivery time equal to $\max_{\boldsymbol{d}} T(\boldsymbol{L}, \boldsymbol{d}, \chi) = \frac{\sum_{r=1}^{\Lambda-\Lambda\gamma} L_{\pi_s(r)}\binom{\Lambda-r}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}}$ which, when averaged over all possible $\boldsymbol{L}$, gives the optimal delay stated in the theorem. Whenever $\gamma \notin \{\frac{1}{\Lambda}, \frac{2}{\Lambda}, \dots, 1\}$ the lower convex envelope of these points is achieved.

[3]We here emphasize that we consider set $S_{\boldsymbol{L}}$ to accept repetitions, i.e., to accept entries consisting of identical vectors. In that sense, note that $|S_{\boldsymbol{L}}| = |S_\Lambda| = \Lambda!$. Repetition will occur whenever there will be caches populated with the same number of users.

Theorem 2. *In the N_0-antenna K-user MISO BC with Λ caches of normalized size γ, the optimal delay for any N_0-admissible class[4] $S_{\boldsymbol{L}}$ is equal to*

$$T^*(S_{\boldsymbol{L}}) = \frac{1}{N_0} Conv\left(\frac{\sum_{r=1}^{\Lambda-\Lambda\gamma} L_{\pi_s(r)} \binom{\Lambda-r}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}}\right) \qquad (4)$$
$$\gamma \in \{\frac{1}{\Lambda}, \frac{2}{\Lambda}, \dots, 1\}$$

which reveals a multiplicative gain of N_0 with respect to the single antenna case.

The lower bound proof is in Section IV, and the scheme is in the long version [17]. An example of the scheme can be found in Section V. The N_0-admissibility condition of Theorem 2 has been relaxed in [1] for a broader range of user-to-cache associations.

As a direct consequence of the above theorem, the following corollary provides, under the assumption of uncoded cache placement, the exact optimal delay for the uniform case[5] $\boldsymbol{L} = (\frac{K}{\Lambda}, \frac{K}{\Lambda}, ..., \frac{K}{\Lambda})$ with $N_0 \leq \frac{K}{\Lambda}$.

Corollary 2. *In the uniform case of $\boldsymbol{L} = (\frac{K}{\Lambda}, \frac{K}{\Lambda}, ..., \frac{K}{\Lambda})$ where $N_0 \leq \frac{K}{\Lambda}$, the optimal delay is*

$$T^*(\boldsymbol{L}) = \frac{K(1-\gamma)}{N_0(\Lambda\gamma+1)}. \qquad (5)$$

The lower bound proof is in Section IV, and the scheme is given in the long version [17].

IV. Derivation of the lower bound

In this section we develop an information theoretic lower bound on the normalized delivery time

$$\mathbb{E}_{\mathbf{L}\in S_{\boldsymbol{L}}}\left(\max_{\mathbf{d}} T(\mathbf{L}, \mathbf{d}, \chi)\right) \qquad (6)$$

associated to any given user-cache association class $S_{\boldsymbol{L}}$. This will directly allow us to prove the results stated in Theorem 1 and Theorem 2. The proof technique is based on the breakthrough in [3] which – for the single stream case with $\Lambda = K$ – employed index coding to bound coded caching performance. Part of the effort in our proof, will be to adapt the index coding approach to account for having shared caches, multiple antennas, and non-uniform user-to-cache association. At this point we note that due to lack of space, the proof is limited to the bare essentials. For a clearer version of the proof, which includes explanatory examples, we refer the reader to [17].

We will start by lower bounding the normalized delivery time $T(\boldsymbol{L}, \boldsymbol{d}, \chi)$, for any user profile $\boldsymbol{L}$, demand vector $\boldsymbol{d}$ and a generic caching-delivery strategy χ. The use of index coding will be facilitated by reordering the demand vector $\mathbf{d}$ to take the form $\mathbf{d}(\mathbf{L}) \triangleq (\mathbf{d}_{\mathcal{U}_1}, \cdots, \mathbf{d}_{\mathcal{U}_\Lambda})$, where $\mathbf{d}_{\mathcal{U}_\lambda}$ is the vector of indices of the files requested by the set of users $\mathcal{U}_\lambda$ associated to cache λ.

The corresponding index coding problem: At this point each requested file $W^{\mathbf{d}_{\mathcal{U}_\lambda(j)}}$ by each user $\mathcal{U}_\lambda(j)$, can be thought to be split into 2^Λ disjoint subfiles $W^{\mathbf{d}_{\mathcal{U}_\lambda(j)}}_{\mathcal{T}}, \mathcal{T} \in 2^{[\Lambda]}$ where $\mathcal{T} \subset [\Lambda]$ indicates the set of helper nodes in which the subfile $W^{\mathbf{d}_{\mathcal{U}_\lambda(j)}}_{\mathcal{T}}$ is cached[6]. Transitioning from the coded caching problem to the equivalent index coding problem, each subfile $W^{\mathbf{d}_{\mathcal{U}_\lambda(j)}}_{\mathcal{T}}$ can be thought to be requested by a user that has as side information all the content $\mathcal{Z}_\lambda$ of helper node λ associated to the original (in the caching problem) user $\mathcal{U}_\lambda(j)$. Naturally no subfile of the form $W^{\mathbf{d}_{U_\lambda(j)}}_{\mathcal{T}}$, $\forall \mathcal{T} \ni \lambda$ is requested because (caching) user $U_\lambda(j)$ already has this information. Therefore the corresponding index coding problem has $K2^{\Lambda-1}$ users and it is represented by the so-called side-information graph $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, where $\mathcal{V}_\mathcal{G}$ is the set of vertices (nodes representing subfiles $W^{\mathbf{d}_{\mathcal{U}_\lambda(j)}}_{\mathcal{T}}, \mathcal{T} \not\ni \lambda$) and $\mathcal{E}_\mathcal{G}$ is the set of direct edges of the graph. A directed edge from node $W^{\mathbf{d}_{\mathcal{U}_\lambda(j)}}_{\mathcal{T}}$ to $W^{\mathbf{d}_{\mathcal{U}_{\lambda'}(j')}}_{\mathcal{T}'}$ exists if and only if $\lambda' \in \mathcal{T}$. This graph $\mathcal{G}$ is defined by $\boldsymbol{L}, \boldsymbol{d}, \chi$, and the total delay T required to serve all index coding users, is our desired $T(\boldsymbol{L}, \boldsymbol{d}, \chi)$. This T is bounded in the following lemma.

Lemma 1. *For the N_0-antenna MISO BC with side information graph $\mathcal{G} = (\mathcal{V}_\mathcal{G}, \mathcal{E}_\mathcal{G})$, then*

$$T \geq \frac{1}{N_0} \sum_{\mathcal{V}\in\mathcal{V}_\mathcal{J}} |\mathcal{V}|$$

holds for every acyclic induced subgraph $\mathcal{J}$ of $\mathcal{G}$, where $\mathcal{V}_\mathcal{J}$ denotes the set of nodes of the subgraph $\mathcal{J}$ and $|\mathcal{V}|$ is the size of the message/subfile $\mathcal{V}$.

The above draws from [18] (see also [19, Corollary1] for a simplified version), and it is easily proved in [17].

Creating large acyclic subgraphs: As we see, the above bound requires the creation of (preferably large) acyclic induced subgraphs of $\mathcal{G}$. The following lemma will tell us how to properly choose a set of nodes that induce a large acyclic subgraph.

Lemma 2. *An induced acyclic subgraph $\mathcal{J}$ of $\mathcal{G}$ corresponding to the index coding problem is designed here to consist of subfiles $W^{\mathbf{d}_{\mathcal{U}_{\pi_s(\lambda)}(j)}}_{\mathcal{T}_\lambda}$, $j \in [L_{\pi_s(\lambda)}]$, $\forall \lambda \in [\Lambda]$ for all $\mathcal{T}_\lambda \subseteq [\Lambda] \setminus \{\pi_s(1), ..., \pi_s(\lambda)\}$ where $\pi_s \in S_\Lambda$ is the permutation such that $L_{\pi_s(1)} \geq L_{\pi_s(2)} \geq ... \geq L_{\pi_s(\Lambda)}$.*

The proof of Lemma 2 is given in the long version [17].

Remark 1. *Lemma 2 is an adaptation of [3, Lemma 1] to our setting. The choice of the permutation π_s is critical; in our case, for each $\boldsymbol{L}, \boldsymbol{d}, \chi$, we pick this single π_s that forces larger (in comparison to other permutations) acyclic subgraphs and thus yields a tighter (eventually optimal) bound. This is different from [3], which instead considered a set of all possible permutations, to ensure a certain symmetry that is crucial to that proof, but which would dilute the non-uniformity in $S_{\boldsymbol{L}}$ that we are capturing here.*

Having chosen an acyclic subgraph according to Lemma 2, we go back to Lemma 1 and form the following lower bound

[4] An $\boldsymbol{L}$ and its class $S_{\boldsymbol{L}}$ are N_0-*admissible* if the following three conditions are met: i) $L_\lambda = \sum_{j=1}^{P} n_{\lambda,j} A_j \quad \forall \lambda \in [\Lambda]$, ii) $A_j \in \mathbb{N}$, $N_0 \leq A_j < 2N_0$ $n_{\lambda,j} \in \mathbb{N}, P \in \mathbb{N}$, and iii) if $n_{\lambda,j} \geq n_{\lambda',j}$ then $n_{\lambda,j'} \geq n_{\lambda',j'}$ $\forall j' \in [P]$. Part of what N_0-admissibility controls is that the number of caches does not exceed a certain threshold. For example, in the uniform case that follows, N_0-admissibility guarantees that $\Lambda \leq \frac{K}{N_0}$.

[5] We here assume that $\Lambda | K$.

[6] Notice that by considering a subpacketization based on the power set $2^{[\Lambda]}$, and by allowing for any possible size of these subfiles, the generality of the result is preserved.

by adding the sizes of all subfiles associated to the chosen acyclic graph.

$$T(\mathbf{L},\mathbf{d},\chi) \geq T^{LB}(\mathbf{L},\mathbf{d},\chi) \triangleq \frac{1}{N_0}\Bigg(\sum_{j=1}^{L_{\pi_s(1)}} \sum_{\mathcal{T}_{\pi_s(1)}\subseteq[\Lambda]\setminus\{\pi_s(1)\}} |W^{\mathbf{d}_{\mathcal{U}_{\pi_s(1)}(j)}}_{\mathcal{T}_{\pi_s(1)}}| + \cdots + \sum_{j=1}^{L_{\pi_s(\Lambda)}} \sum_{\mathcal{T}_{\pi_s(\Lambda)}\subseteq[\Lambda]\setminus\{\pi_s(1),\ldots,\pi_s(\Lambda)\}} |W^{\mathbf{d}_{\mathcal{U}_{\pi_s(\Lambda)}(j)}}_{\mathcal{T}_{\pi_s(\Lambda)}}| \Bigg). \tag{7}$$

At this point we average over $\boldsymbol{L}\in S_{\boldsymbol{L}}$ to get

$$T(S_{\boldsymbol{L}},\chi) \triangleq \mathbb{E}_{\boldsymbol{L}\in S_{\boldsymbol{L}}}\left(\max_{\mathbf{d}} T(\boldsymbol{L},\mathbf{d},\chi)\right) \overset{(a)}{\geq} \frac{1}{|\mathcal{D}_{wc}|}\frac{1}{|S_{\boldsymbol{L}}|}\sum_{\mathbf{d}\in\mathcal{D}_{wc}}\sum_{\boldsymbol{L}\in S_{\boldsymbol{L}}} T(\mathbf{L},\mathbf{d},\chi) \tag{8}$$

which yields

$$T(S_{\boldsymbol{L}},\chi) \overset{(b)}{\geq} \frac{1}{P(N,K)\Lambda!}\sum_{\mathbf{d}\in\mathcal{D}_{wc}}\sum_{\boldsymbol{L}\in S_{\boldsymbol{L}}} T^{LB}(\mathbf{L},\mathbf{d},\chi) \tag{9}$$

where in the above, (a) is due to interchanging the max and average operation and due to the worst-case delay assumption where each $\mathbf{d}$ belongs to set $\mathcal{D}_{wc}$ consisting of requests vectors with K different files. Finally (b) is due to having $|\mathcal{D}_{wc}| = P(N,K)$, $|S_{\boldsymbol{L}}| = \Lambda!$, and due to combining (7) and (8).

After applying (7), we can rewrite the double summation in (9), to get

$$\sum_{\mathbf{d}\in\mathcal{D}_{wc}}\sum_{\boldsymbol{L}\in S_{\boldsymbol{L}}} T^{LB}(\mathbf{L},\mathbf{d},\chi) = \tag{10}$$
$$\frac{1}{N_0}\sum_{i=0}^{\Lambda}\sum_{n\in[N]}\sum_{\mathcal{T}\subseteq[\Lambda]:|\mathcal{T}|=i}|W^n_{\mathcal{T}}|\cdot \underbrace{\sum_{\mathbf{d}\in\mathcal{D}_{wc}}\sum_{\boldsymbol{L}\in S_{\boldsymbol{L}}}\mathbb{1}_{\mathcal{V}_{\mathcal{J}_s^{\mathbf{d}(\boldsymbol{L})}}}(W^n_{\mathcal{T}})}_{Q_i(W^n_{\mathcal{T}})}$$

where $\mathcal{V}_{\mathcal{J}_s^{\mathbf{d}(\boldsymbol{L})}}$ is the set of vertices in the acyclic subgraph chosen according to Lemma 2 for a given $\mathbf{d}$ and $\mathbf{L}$.

A crucial step towards removing the dependence on $\mathcal{T}$, comes from the fact that

$$Q_i = Q_i(W^n_{\mathcal{T}}) \triangleq \sum_{\mathbf{d}\in\mathcal{D}_{wc}}\sum_{\boldsymbol{L}\in S_{\boldsymbol{L}}}\mathbb{1}_{\mathcal{V}_{\mathcal{J}_s^{\mathbf{d}(\boldsymbol{L})}}}(W^n_{\mathcal{T}}) = \binom{N-1}{K-1}\sum_{r=1}^{\Lambda} P(\Lambda-i-1,r-1)(\Lambda-r)!L_{\pi_s(r)} \times P(K-1,L_{\pi_s(r)}-1)(K-L_{\pi_s(r)})!(\Lambda-i) \tag{11}$$

where we notice that the total number of times that a specific subfile appears — in the summation in (10), over the set of all possible $\mathbf{d}\in\mathcal{D}_{wc}$, $\boldsymbol{L}\in S_{\boldsymbol{L}}$, and given the chosen permutation π_s — is not dependent on the subfile itself but is dependent only on the number of caches $i = |\mathcal{T}|$ storing that subfile. The proof of (11) can be found in [17].

By considering $x_i \triangleq \sum_{n\in[N]}\sum_{\mathcal{T}\subseteq[\Lambda]:|\mathcal{T}|=i}|W^n_{\mathcal{T}}|$ to be the total amount of data stored in exactly i helper nodes, and by noting that

$$N = \sum_{i=0}^{\Lambda} x_i = \sum_{i=0}^{\Lambda}\sum_{n\in[N]}\sum_{\mathcal{T}\subseteq[\Lambda]:|\mathcal{T}|=i}|W^n_{\mathcal{T}}| \tag{12}$$

we can combine (9), (10) and (11) to get

$$T(S_{\boldsymbol{L}},\chi) \geq \frac{1}{N_0}\sum_{i=0}^{\Lambda}\frac{Q_i}{P(N,K)\Lambda!}x_i. \tag{13}$$

Now applying (11), after some manipulation, we get

$$T(S_{\boldsymbol{L}},\chi) \geq \frac{1}{N_0}\sum_{i=0}^{\Lambda}\frac{\sum_{r=1}^{\Lambda-i}L_{\pi_s(r)}\binom{\Lambda-r}{i}}{N\binom{\Lambda}{i}}x_i = \frac{1}{N_0}\sum_{i=0}^{\Lambda}\frac{x_i}{N}c_i \tag{14}$$

where $c_i \triangleq \frac{\sum_{r=1}^{\Lambda-i}L_{\pi_s(r)}\binom{\Lambda-r}{i}}{\binom{\Lambda}{i}}$ decreases with $i\in\{0,1,...,\Lambda\}$ (see [17]).

Under the file-size and cache-size constraints $\sum_{i=0}^{\Lambda}x_i = N$, $\sum_{i=0}^{\Lambda} i x_i \leq KM$ respectively, (14) gives a lower bound on the delay of any caching-and-delivery scheme χ whose caching policy implies a set of $\{x_i\}$. We then employ the Jensen's-inequality based technique of [4, Proof of Lemma 2] to minimize the above, over all admissible $\{x_i\}$. First we see that for any integer $\Lambda\gamma$, we get that

$$T(S_{\boldsymbol{L}},\chi) \geq \frac{1}{N_0}\frac{\sum_{r=1}^{\Lambda-\Lambda\gamma}L_{\pi_s(r)}\binom{\Lambda-r}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}} \tag{15}$$

and for all other values of $\Lambda\gamma$, this is extended to its convex lower envelop. The details of deriving (15) are found in [17].

The above concludes lower bounding $\mathbb{E}_{\mathbf{L}\in S_{\boldsymbol{L}}}(\max_{\mathbf{d}} T(\mathbf{L},\mathbf{d},\chi))$, for any scheme χ. Hence the above automatically concludes the proof of the lower bound part of Corollary 1, as well as the proof of the lower bound part for Theorem 2, yielding $T^*(S_{\boldsymbol{L}}) \geq \frac{1}{N_0}Conv\left(\frac{\sum_{r=1}^{\Lambda-\Lambda\gamma}L_{\pi_s(r)}\binom{\Lambda-r}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}}\right)$, for $\gamma\in\{\frac{1}{\Lambda},\frac{2}{\Lambda},\ldots,1\}$ which holds[7] for any $S_{\boldsymbol{L}}$.

For Corollary 2, we simply consider the uniform $S_{\boldsymbol{L}}$ and then apply Pascal's triangle, while for Theorem 1, we directly have that

$$T^* \geq \mathbb{E}_{S_{\boldsymbol{L}}}Conv\left(\frac{\sum_{r=1}^{\Lambda-\Lambda\gamma}L_{\pi_s(r)}\binom{\Lambda-r}{\Lambda\gamma}}{\binom{\Lambda}{\Lambda\gamma}}\right). \quad \square$$

V. Example of scheme

Due to lack of space, we describe the scheme for a specific example case. The reader is referred to [17] for the general description of the scheme and is encouraged to read [1] for an improved version of the algorithm. Let $K=15, N_0=2, N=15$, and consider $\Lambda=3$ helper caches of size $M=5$ units of file. We first split each file W^n in 3 equal parts W^n_1, W^n_2, W^n_3, and as in [2], each cache λ stores $W^n_\lambda,\ \forall n\in[15]$.

Let users $\mathcal{U}_1=\{1,2,...,8\}$ be associated to helper node 1, users $\mathcal{U}_2=\{9,10,...,13\}$ to helper node 2, and users $\mathcal{U}_3=\{14,15\}$ to helper node 3. This association implies $\boldsymbol{L}=(8,5,2)$. Let us assume the demand vector $\mathbf{d}=(1,2,...,15)$.

The proposed optimal scheme will consider two rounds, with the first one serving users $\mathcal{R}_1=\{1,2,9,10,14,15\}$, and the second serving users $\mathcal{R}_2=\{3,4,5,6,7,8,11,12,13\}$. In the first round, the 3 transmissions are:

$$\boldsymbol{x}_{\{1,2,9,10\}} = \mathbf{H}^{-1}_{\{1,2\}}\begin{bmatrix}W^1_2\\W^2_2\end{bmatrix} + \mathbf{H}^{-1}_{\{9,10\}}\begin{bmatrix}W^9_1\\W^{10}_1\end{bmatrix} \tag{16}$$

$$\boldsymbol{x}_{\{1,2,14,15\}} = \mathbf{H}^{-1}_{\{1,2\}}\begin{bmatrix}W^1_3\\W^2_3\end{bmatrix} + \mathbf{H}^{-1}_{\{14,15\}}\begin{bmatrix}W^{14}_1\\W^{15}_1\end{bmatrix} \tag{17}$$

$$\boldsymbol{x}_{\{9,10,14,15\}} = \mathbf{H}^{-1}_{\{9,10\}}\begin{bmatrix}W^9_3\\W^{10}_3\end{bmatrix} + \mathbf{H}^{-1}_{\{14,15\}}\begin{bmatrix}W^{14}_2\\W^{15}_2\end{bmatrix} \tag{18}$$

[7]The outer bound does not need the N_0-admissibility condition.

where $\mathbf{H}_{\{i,j\}}^{-1}$ is the zero-forcing (ZF) precoder for the channel $\mathbf{H}_{\{i,j\}} = [\mathbf{h}_i^T \mathbf{h}_j^T]$ to users i and j. Hence user 1, during the first transmission, receives $y_1 = W_2^1 + [1\ 0]\mathbf{h}_1^T \mathbf{H}_{\{9,10\}}^{-1} \begin{bmatrix} W_1^9 \\ W_1^{10} \end{bmatrix} + w_1$ and simply caches-out W_1^9 and W_1^{10} to decode W_2^1. Similarly for the other users.

In the second round, each remaining subfile is split into two parts as $W_{\mathcal{T}}^n = \{W_{\mathcal{T},1}^n, W_{\mathcal{T},2}^n\}$. This round is split into two sub-rounds, where the first sub-round serves users $3,4,5,11,12,13$ and the second serves users $6,7,8$. The first 3 transmissions for the first sub-round are

$$\begin{aligned}
\boldsymbol{x}_{\{3,4,11,12\}} =& \mathbf{H}_{\{3,4\}}^{-1} \begin{bmatrix} W_{2,1}^3 \\ W_{2,1}^4 \end{bmatrix} + \mathbf{H}_{\{11,12\}}^{-1} \begin{bmatrix} W_{1,1}^{11} \\ W_{1,1}^{12} \end{bmatrix} \\
\boldsymbol{x}_{\{3,5,11,13\}} =& \mathbf{H}_{\{3,5\}}^{-1} \begin{bmatrix} W_{2,2}^3 \\ W_{2,1}^5 \end{bmatrix} + \mathbf{H}_{\{11,13\}}^{-1} \begin{bmatrix} W_{1,2}^{11} \\ W_{1,1}^{13} \end{bmatrix} \\
\boldsymbol{x}_{\{4,5,12,13\}} =& \mathbf{H}_{\{4,5\}}^{-1} \begin{bmatrix} W_{2,2}^4 \\ W_{2,2}^5 \end{bmatrix} + \mathbf{H}_{\{12,13\}}^{-1} \begin{bmatrix} W_{1,2}^{12} \\ W_{1,2}^{13} \end{bmatrix}
\end{aligned}$$

each serving 4 users, while the rest, as seen below, are each intended for 2 users.

$$\begin{aligned}
\boldsymbol{x}_{\{3,4\}} &= \mathbf{H}_{\{3,4\}}^{-1} \begin{bmatrix} W_{3,1}^3 \\ W_{3,1}^4 \end{bmatrix} \quad \boldsymbol{x}_{\{3,5\}} = \mathbf{H}_{\{3,5\}}^{-1} \begin{bmatrix} W_{3,2}^3 \\ W_{3,1}^5 \end{bmatrix} \\
\boldsymbol{x}_{\{4,5\}} &= \mathbf{H}_{\{4,5\}}^{-1} \begin{bmatrix} W_{3,2}^4 \\ W_{3,2}^5 \end{bmatrix} \quad \boldsymbol{x}_{\{11,12\}} = \mathbf{H}_{\{11,12\}}^{-1} \begin{bmatrix} W_{3,1}^{11} \\ W_{3,1}^{12} \end{bmatrix} \\
\boldsymbol{x}_{\{11,13\}} &= \mathbf{H}_{\{11,13\}}^{-1} \begin{bmatrix} W_{3,2}^{11} \\ W_{3,1}^{13} \end{bmatrix} \quad \boldsymbol{x}_{\{12,13\}} = \mathbf{H}_{\{12,13\}}^{-1} \begin{bmatrix} W_{3,2}^{12} \\ W_{3,2}^{13} \end{bmatrix}
\end{aligned}$$

We can now easily verify that the intended users $3,4,5,11,12,13$ can successfully decode. The last sub-round serves users $6,7,8$. These users cannot benefit from coded multicasting because they share the same cache content, hence ZF is applied as follows:

$$\begin{aligned}
\boldsymbol{x}_{\{6,7\}} =& \mathbf{H}_{\{6,7\}}^{-1} \begin{bmatrix} W_{2,1}^6 | W_{3,1}^6 \\ W_{2,1}^7 | W_{3,1}^7 \end{bmatrix}, \ \boldsymbol{x}_{\{6,8\}} = \mathbf{H}_{\{6,8\}}^{-1} \begin{bmatrix} W_{2,2}^6 | W_{3,2}^6 \\ W_{2,1}^8 | W_{3,1}^8 \end{bmatrix} \\
\boldsymbol{x}_{\{7,8\}} =& \mathbf{H}_{\{7,8\}}^{-1} \begin{bmatrix} W_{2,2}^7 | W_{3,2}^7 \\ W_{2,2}^8 | W_{3,2}^8 \end{bmatrix}.
\end{aligned}$$

The delay $T = \frac{1}{3}\cdot 3 + \frac{1}{6}\cdot 9 + \frac{1}{6}\cdot 6 = \frac{21}{6}$, matches $T^*(S_{\{8,5,2\}}) \geq \frac{\sum_{r=1}^{2} L_{\pi_S(r)} \binom{3-r}{1}}{2\binom{3}{1}} = \frac{8\cdot 2 + 5\cdot 1}{6} = \frac{21}{6}$ from Theorem 2.

VI. Conclusions

The work further bridges the gap between realistic wireless networks and coded caching, and is among the first to enlist index coding as a means of providing (in this case, exact) outer bounds for more involved cache-aided network topologies that better capture aspects of cache-aided wireless networks, such as shared caches and user-cache association non-uniformities. These non-uniformities raised an interesting challenge in redesigning outer bounds, as well as redesigning coded caching which is generally known to thrive on symmetry. As we have shown in this work, the non-uniformity of user-to-cache association results in reduced DoF compared to the uniform setting; the higher is the skewness of the users' distribution among the caches, the lower is the optimal achievable DoF.

A multiplicative relationship between caching gain and multiplexing gain: One important conclusion is on the interplay between the number of antennas and the number of different caches. Focusing here on the uniform case where each cache serves K/Λ users, this work revealed that as long as $N_0 \leq K/\Lambda$, the derived DoF is $d_\Sigma = N_0(1+\Lambda\gamma)$ (users served at a time), thus revealing the powerful impact of adding antennas; for example introducing a second transmit antenna can double the DoF. This multiplicative effect comes in strong contrast to the additive effect experienced in the standard cache-aided BC setting where all $K = \Lambda$ users have their own cache, in which case, as we know from [20], adding one antenna allows for one additional DoF.

Similarly, again assuming that $\Lambda \leq K/N_0$, directly tells us that every time we add a single degree of cache-redundancy (i.e., every time we increase $\Lambda\gamma$ by one), we gain N_0 degrees of freedom. This is again in contrast to the case of $\Lambda = K$, where again a unit increase in the cache redundancy yields only one additional DoF. The above observations are further evidence of the powerful impact of jointly introducing a modest number of antennas and a modest number of helper nodes.

References

[1] E. Parrinello, A. Unsal, and P. Elia, "Optimal Coded Caching in Heterogeneous Networks with Uncoded Prefetching," *arXiv preprint www.arxiv.org*, 2018.

[2] M. Maddah-Ali and U. Niesen, "Fundamental Limits of Caching," *IEEE Transactions on Information Theory*, May 2014.

[3] K. Wan, D. Tuninetti, and P. Piantanida, "On the Optimality of Uncoded Cache Placement," in *Information Theory Workshop (ITW), 2016 IEEE*, 2016.

[4] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The Exact Rate-Memory Tradeoff for Caching with Uncoded Prefetching," *IEEE Transactions on Information Theory*, Feb 2017.

[5] J. Zhang and P. Elia, "Wireless Coded Caching: A Topological Perspective," in *IEEE International Symposium on Information Theory, (ISIT)*, 2017.

[6] S. S. Bidokhti, M. Wigger, and R. Timo, "Erasure Broadcast Networks with Receiver Caching," in *IEEE International Symposium on Information Theory, (ISIT)*, 2016.

[7] A. Ghorbel, M. Kobayashi, and S. Yang, "Cache-Enabled Broadcast Packet Erasure Channels with State Feedback," in *Proc. Allerton Conference*, 2015.

[8] J. Zhang and P. Elia, "Fundamental Limits of Cache-Aided Wireless BC: Interplay of Coded-Caching and CSIT Feedback," *IEEE Transactions on Information Theory*, May 2017.

[9] A. Sengupta, R. Tandon, and O. Simeone, "Cache Aided Wireless Networks: Tradeoffs Between Storage and Latency," in *Conference on Information Sciences and Systems, CISS*, 2016.

[10] Y. Cao, M. Tao, F. Xu, and K. Liu, "Fundamental Storage-Latency Tradeoff in Cache-Aided MIMO Interference Networks," *IEEE Transactions on Wireless Communications*, 2017.

[11] J. S. P. Roig, D. Gündüz, and F. Tosato, "Interference Networks with Caches at Both Ends," in *2017 IEEE International Conference on Communications (ICC)*, May 2017.

[12] E. Piovano, H. Joudeh, and B. Clerckx, "On coded Caching in the Overloaded miso Broadcast Channel," in *2017 IEEE International Symposium on Information Theory (ISIT)*, June 2017.

[13] E. Lampiris and P. Elia, "Achieving Full Multiplexing and Unbounded Caching Gains with Bounded Feedback Resources," in *2018 IEEE International Symposium on Information Theory (ISIT)*, June 2018.

[14] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless Video Content Delivery through Distributed Caching Helpers," in *INFOCOM*, March 2012.

[15] J. Hachem, N. Karamchandani, and S. Diggavi, "Coded Caching for Multi-Level Popularity and Access," *IEEE Transactions on Information Theory*, May 2017.

[16] M. A. Maddah-Ali and U. Niesen, "Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff," *IEEE/ACM Transactions on Networking*, 2015.

[17] E. Parrinello, A. Ünsal, and P. Elia, "Coded Caching in Heterogeneous Networks with Uncoded Prefetching," 2018. [Online]. Available: http://www.eurecom.fr/~elia/pubs/ITWlong2018.pdf

[18] P. Sadeghi, F. Arbabjolfaei, and Y. H. Kim, "Distributed Index Coding," in *2016 IEEE Information Theory Workshop, ITW*, 2016.

[19] M. Li, L. Ong, and S. J. Johnson, "Cooperative Multi-Sender Index Coding," *arXiv preprint arXiv:1701.03877*, 2017.

[20] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-Server Coded Caching," *IEEE Transactions on Information Theory*, Dec 2016.